



Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires

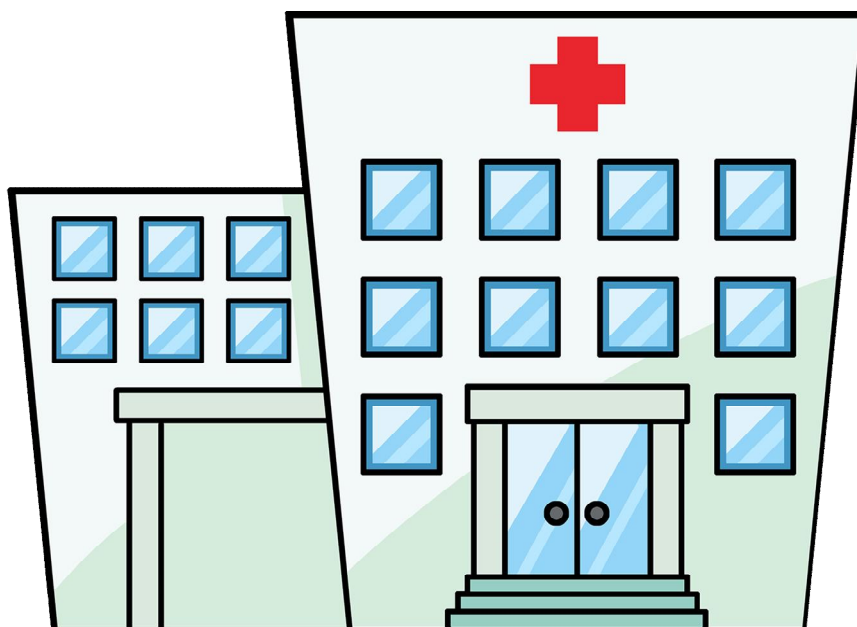
# Gestión de Datos

Trabajo Práctico

2° Cuatrimestre 2013

CLINICA – FRBA

Enunciado V1.0



## Índice

Introducción.....	3
Objetivos generales.....	3
Descripción general .....	3
Componentes del TP.....	4
Base de Datos y Modelo de Datos .....	4
Aplicación Desktop.....	4
Requerimientos.....	4
General .....	4
Base de Datos .....	5
Aplicación Desktop.....	5
General .....	14
Base de Datos .....	14
Aplicación Desktop.....	15
Restricciones de la solución .....	17
Condiciones de aprobación .....	17
Testing.....	17
Modelo de Datos.....	18
Consultas SQL .....	18
Respetar Guía de ABMs.....	18
Aplicación Desktop.....	18
Fecha de entrega y condiciones .....	18
Sobre los grupos .....	19
Ayuda y contacto .....	19
Donde aprender C#.....	20
Sobre la elección de C#.....	20
Obtención de herramientas .....	21
Formato de entrega .....	22
Lugar de envío .....	22
Estructura del archivo zip.....	23
Readme.txt:.....	23
Estrategia.pdf:.....	23
\src:.....	24
\data:.....	24

## **Introducción**

### **Objetivos generales**

El presente trabajo práctico persigue los siguientes objetivos generales

- Promover la investigación de técnicas de base de datos.
- Aplicar la teoría vista en la asignatura en una aplicación concreta.
- Desarrollar y probar distintos algoritmos sobre datos reales.
- Utilizar un lenguaje de programación de última generación.
- Fomentar la delegación y el trabajo en grupo.

### **Descripción general**

Mediante este trabajo práctico se intenta simular la implementación de nuevo sistema del ramo de la medicina.

Abarca la problemática de los afiliados y la implementación de un nuevo módulo que le permita comprar bonos desde cualquier pc, como así también el pedido de turnos para los diferentes profesionales que conforman el staff de la clínica.

Se disponen de bonos para consulta y bonos para farmacia, los cuales varían su precio en función del plan médico que dispone el afiliado, dicho precio debe ser parametrizado y modificado en función del tiempo.

Por cuestión de tiempo de elaboración y duración de la materia solo se llegará a la generación de la receta digital médica y quedará para una segunda etapa los módulos de facturación de medicamentos.

La implementación de dicho sistema, requiere previamente realizar la migración de los datos que se tenían hasta el momento.

Es para ello que se necesita que se reformulen los procesos y el diseño de la base de datos que cumpla con las nuevas restricciones y se adecue a la situación actual.

## **Componentes del TP**

El alumno recibirá dos componentes ya hechos del sistema y, en base a estos deberá crear uno nuevo e implementar nuevas funcionalidades. Los componentes a recibir son:

### **Base de Datos y Modelo de Datos**

La cátedra provee un script que permite crear una base de datos en el motor SQL Server 2008. Esta base de datos incluye el modelo de una única tabla, llamada maestra, que es cargada con datos provistos por la cátedra. Los datos de esa tabla se encuentran desorganizados y no poseen ningún tipo de normalización.

El alumno deberá estudiar los datos recibidos y confeccionar un modelo de datos que siga todos los standards de desarrollo de bases de datos explicados durante la cursada.

Los datos de esta tabla maestra pertenecen a un dominio de gestión de turnos, consultas y compras de bonos.

El sistema a desarrollar será utilizado por 2 tipos de usuarios distintos: administradores y clientes.

Parte de la lógica del negocio a resolver deberá ser inferida por el alumno, en base a las columnas y valores presentes en los datos. De todas maneras es recomendable consultar al grupo de Google de la materia antes de tomar decisiones incorrectas.

### **Aplicación Desktop**

La cátedra provee un proyecto C# a modo de template, sobre el cual deberá desarrollarse una aplicación Desktop que interactúe con la nueva base de datos, cuyo diseño estará a cargo de los alumnos. La aplicación deberá ser del tipo Desktop desarrollada sobre C# con Visual Studio 2008 y Framework de .NET 3.5.

Esta aplicación tendrá diversas pantallas, reportes y formularios que permitirán interactuar, cargar y visualizar la información de la base de datos de SQL Server.

## **Requerimientos**

### **General**

El alumno deberá crear todos los componentes de base de datos e implementar todas las funcionalidades pedidas para la aplicación Desktop, cumpliendo con las siguientes pautas:

## Base de Datos

El alumno deberá crear un modelo de datos que **organice y normalice** los datos de la única tabla provista por la cátedra. Este modelo de datos incluye:

- Creación de nuevas tablas y vistas.
- Creación de claves primarias y foráneas para relacionar estas tablas.
- Creación de constraints y triggers sobre estas tablas cuando fuese necesario.
- Creación de los índices para acceder a los datos de estas tablas de manera eficiente.
- Migración de datos: Cargar todas las tablas creadas utilizando la totalidad de los datos entregados por la cátedra en la única tabla del modelo. Para este punto deberán utilizarse Stored Procedures sobre la base de datos. No podrá realizarse la migración de datos utilizando la aplicación Desktop ni ninguna otra herramienta auxiliar.

El alumno deberá entregar un único archivo de Script que al ejecutar realice todos los pasos mencionados anteriormente, en el orden correcto. Todo el modelo de datos confeccionado por el alumno deberá ser creado y cargado correctamente ejecutando este Script una única vez, antes de empezar a testear la aplicación Desktop.

Todas las columnas creadas para las nuevas tablas **deberán respetar los mismos tipos de datos** de las columnas existentes en la tabla principal. A su vez el alumno podrá crear nuevas columnas, claves e identificadores para satisfacer sus necesidades. Pero nunca se podrá inventar información, por ejemplo crear un afiliado que nunca existió.

## Aplicación Desktop

El alumno deberá crear una aplicación Desktop en C# sobre Visual Studio 2008 con Framework .NET versión 3.5. Esta aplicación deberá contar con formularios, reportes y tablas. Todos estos componentes deberán respetar los lineamientos planteados en el documento “Guía de ABMs”. Es recomendable leerlo en este punto antes de continuar con el enunciado.

En su primera fase de prototipo el sistema contará con el siguiente listado que se detalla a continuación:

1. [Abm de Rol.](#)
2. [Login y seguridad.](#)
3. [Registro de Usuario.](#)
4. [Abm Afiliado.](#)
5. [Abm Profesional.](#)
6. [Abm Especialidades Médicas.](#)
7. [Abm de Planes.](#)
8. [Registrar agenda del médico.](#)
9. [Compra de bonos.](#)
10. [Pedir turno.](#)
11. [Registro de llegada para atención médica.](#)
12. [Registrar resultado para atención médica.](#)
13. [Cancelar atención médica.](#)
  - De parte del médico.

- De parte del paciente.
- 14. [Generar Receta](#)
- 15. [Listado estadístico](#)

El listado total de funcionalidades del sistema es fijo y no varía.

La funcionalidad de la aplicación deberá responder a los siguientes requerimientos de negocio:

## 1. [ABM de Rol](#)

Funcionalidad para poder crear, modificar y eliminar el acceso de un usuario a una opción del sistema.

Crear un rol implica cargar los siguientes datos:

- Nombre
- Listado de Funcionalidades (selección acotada)

Todos los datos mencionados anteriormente son obligatorios.

Un rol posee un conjunto de funcionalidades y las mismas no pueden estar repetidas dentro de un rol en particular.

Debe tenerse en cuenta, que actualmente existen 3 roles.

- Afiliado
- Administrativo
- Profesional

En la modificación de un rol solo se pueden alterar ambos campos: el nombre y el listado de funcionalidades. Se deben poder quitar de a una las funcionalidades como así también agregar nuevas funcionalidades a rol que se está modificando.

La eliminación del rol implica una baja lógica del mismo. El rol debe poder inhabilitarse. No permitido la asignación de un rol inhabilitado a un usuario, por ende, se le debe quitar el rol inhabilitado a todos aquellos usuarios que lo posean.

Se debe poder volver a habilitar un rol inhabilitado desde la sección de modificación. Esto no implica recuperar las asignaciones que existían en un pasado.

Para elegir el rol que se desea modificar o eliminar se debe mostrar un listado con todos los roles existentes en el sistema.

## 2. [Login y Seguridad](#)

Al ejecutar la aplicación el usuario no podrá acceder a ninguna funcionalidad del sistema hasta completar el proceso de Login.

El proceso de Login pedirá al usuario su Username y su Password. Si Login es correcto, el usuario podrá acceder al sistema, EN CASO DE TENER ASIGNADO MAS DE UN ROL, SE DEBERÁ SELECCIONAR CON CUAL DE ELLOS SE QUIERE LOGUEAR. Al acceder al mismo, la aplicación solo deberá generar y mostrar las entradas de menú disponibles para este usuario, según los roles del mismo. El usuario no debe ni siquiera ver las funcionalidades a las que no posee acceso.

Si el Login es incorrecto el usuario no podrá acceder al sistema. Se debe volver a mostrar el Login para que intente nuevamente. El sistema debe llevar un registro de

cantidad intentos fallidos de login. Luego de 3 intentos fallidos en cualquier momento, el usuario debe ser inhabilitado. Al realizar un Login satisfactorio, el sistema deberá limpiar la cantidad de intentos fallidos.

El Login se considera una funcionalidad de características especiales. No se considera una funcionalidad que puede ser asignada a un rol. Todos los usuarios tienen la capacidad de utilizar el Login.

### 3. [Registro de Usuario](#)

Para reducir el tiempo de confección del TP y así beneficiar al alumno se determinó que no será necesario que se implemente/codifique el ABM de usuarios. Pero deberán tener en cuenta que al momento de realizar la entrega de dicho TP, deben entregar un usuario con perfil Administrador, username admin y la password a asignar será **w23e**, para el resto de los usuarios (afiliados y profesionales) los alumnos determinarán cuáles serán los username y password. ESTO NO JUSTIFICA BAJO NINGUNA CIRCUNSTANCIA LA NO MODELACIÓN DE LA ENTIDAD USUARIO.

El username debe ser único en un todo el sistema.

El password deberá almacenarse encriptado de forma irreversible bajo el algoritmo de encriptación SHA256.

### 4. [ABM de Afiliados](#)

Funcionalidad que permite a un administrativo del hospital crear, modificar y dar de baja un afiliado. Se entiende por paciente a toda aquella persona que puede ser atendida en el hospital.

Es de suponer que un paciente a lo largo de su historia puede sufrir modificaciones en alguno de sus datos, como ser su dirección, teléfono, mail, plan médico, etc.( no así su nombre, apellido, dni y fecha de nacimiento) Si fuese necesario modificar el plan del afiliado, ese necesario que se registre cuando se ha producido dicha modificación y el motivo que la originó, de manera de poder obtener un historial de dichos cambios.

En caso de querer dar de baja un afiliado, la misma será por medio de baja lógica y registrando la fecha de dicho acontecimiento, si el paciente tenía turnos registrados o recetas emitidas éstas no podrán ser utilizadas dado que dicho paciente se ha desvinculado de la entidad.

Los datos fundamentales a registrar son los siguientes.

- Nombre y Apellido.
- Tipo y número de documento
- Dirección completa
- Teléfono
- mail
- Fecha de nacimiento
- Sexo

- Estado civil
  - Soltero/a
  - Casado/a
  - Viudo/a
  - Concubinato
  - Divorciado/a
- Cantidad de hijos o familiares a cargo
- Plan Médico
- Nro. de afiliado (asignado por el sistema)

Antes de finalizar la operación y si el futuro nuevo afiliado es casado/a o vive en concubinato se tendrá que dar la opción de asociar a su conyugue y se deberán cargar todo los datos necesarios para su registro.

Independientemente de lo expuesto en el párrafo anterior, se deberá consultar si el nuevo afiliado tiene hijos o familiares a cargo. En caso de que así sea, se deberá dar la posibilidad de ingresar los datos necesarios para el registro, ya que está en todo su derecho de querer afiliarlos o no, mismo caso para el conyugue.

Una vez completado todos los datos de registro, el sistema procederá a calcular el número de afiliado para el nuevo usuario y si fuese necesario para el resto del grupo familiar, para ello, el afiliado principal (el que generó el registro principalmente) será registrado con un determinado número y finalizará con 01, para el conyugue 02 y 03 y subsiguientes para los hijos o familiares a cargo según corresponda. Por ende todo el grupo familiar compartirá el mismo número raíz de afiliado pero se identificarán uno de otro en función de los últimos 2 dígitos. El número de afiliado es NUMERO, no es necesario separarlos con guiones, barras, o cualquier otro carácter para marcar la división de la raíz y los 2 dígitos siguientes.

Las modificaciones de los afiliados se tendrán que realizar en forma independiente, no será necesario cargar toda la estructura familiar por ej si se quiere modificar algún dato de un hijo o familiar a cargo.

## 5. [ABM de Profesional](#)

Esta funcionalidad es la que permite a un administrativo crear, modificar, o dar de baja los diferentes profesionales que atienden en el hospital.

Un profesional puede tener asociado una o más especialidades en la cual se puede desempeñar.

Los datos fundamentales a registrar son los siguientes.

- Nombre y Apellido.
- Tipo y número de documento
- Dirección completa
- Teléfono
- Mail
- Fecha de nacimiento
- Sexo
- Matricula.
- Especialidades.



Si fuese necesario, dar de baja a algún profesional, la misma será por medio de baja lógica y además se tendrán que dar de baja todos los turnos de los diferentes pacientes que tiene asignado como así también no estar más disponible para futuras consultas.

#### 6. [ABM de Especialidades Médicas](#)

En esta funcionalidad se detallan cada una de las especialidades médicas en las cuales se desempeñan cada uno de los profesionales que cuenta el Hospital.

Las especialidades se caracterizan por tener una descripción que las identifica y un tipo de especialidad, por ejemplo. Cirugía Cardiovascular como especialidad y su tipo de especialidad sería especialidades quirúrgicas.

Para reducir el tiempo de desarrollo de los alumnos, no será necesario que realicen la implementación de dicho ABM.

#### 7. [ABM de Plan](#)

Esta funcionalidad hace hincapié a los diferentes planes que admite la clínica.

El plan básicamente consiste en el nivel de servicio que la clínica brindará a sus afiliados (en función de la cuota que el mismo abone), además de lo anteriormente expuesto se expresa en el plan el precio del bono que el afiliado deberá comprar para realizar las consultas o en el expendio de los medicamentos.

En función del tipo de plan se determinará el precio de los mismos, por ej. para 2 planes distintos el bono de consulta no tiene el mismo precio.

Como la funcionalidad fue simplificada no es necesario realizar la implementación de este caso de uso pero si su modelado.

#### 8. [Registrar Agenda Profesional.](#)

Cada profesional que atiende en la clínica tiene la necesidad de llevar una agenda.

En dicha agenda se registrará los días en los cuales estarán disponibles los diferentes profesionales que conforman la cartilla de la clínica. Para ello será necesario determinar los días en los cuales el profesional atiende, pudiendo seleccionar algunos o todos dentro del rango de Lunes a Sábados. Luego de ello será necesario establecer el rango horario de atención, teniendo en cuenta que la clínica establece su horario de atención desde las 7:00 hasta las 20:00 para los días hábiles y desde las 10:00 hasta las 15:00 para los días sábados

Una vez determinado el profesional, los días de atención y el rango horario (con turnos disponibles cada 30 minutos), será necesario que se termine el rango de fechas para el cual estará disponible.

Otro factor muy importante a tener en cuenta es que un profesional no puede tener acumuladas más de 48 horas laborales por semana y no necesariamente tener la misma carga horaria en los días que presta servicio.

El período para la confección de la agenda de un profesional no puede ser superior a los 120 días corridos.

No se tendrán en cuenta para este TP realizar modificaciones en las agendas ya asignadas, en caso de error de carga se utilizará la opción de cancelar turno.

## 9. Compra de Bonos.

Este caso de uso detalla el proceso de compra de los bonos por parte del usuario. Dentro del sistema se encuentran 2 tipos de bonos:

- Bono consulta
- Bono farmacia

Dichos bonos varían su precio en función del plan al que pertenezcan.

Los bonos consulta son aquellos utilizados por los afiliados para la atención con algún profesional de la clínica, y un bono solo es válido para una atención.

Los bonos farmacia son aquellos que utiliza el afiliado en la compra de medicamentos recetados por los médicos de la clínica, en función de la atención médica prestada por dicho profesional.

Estos bonos, una vez comprados, se les asigna una fecha de compra, que para el caso de los bonos farmacia es utilizada para calcular la fecha de vencimiento y la misma es de 60 días y debe imprimirse en el bono. Por su parte, los bonos consulta tienen un campo “numero de consulta”, propio de cada afiliado, el mismo es completado en el momento que se efectiviza la consulta médica en forma parcial (cuando el afiliado registra su llegada a la atención) que indica la cantidad de veces que el afiliado hizo una consulta médica independientemente del especialista que lo atiende.

Una compra puede conllevar a la adquisición tanto de bonos consulta como bonos farmacia sin límite.

Luego de finalizada la operación de compra se deberá informar el monto a pagar por los bonos comprados quedando registrado en el sistema dicha suma y la cantidad comprada, junto con el afiliado que realizó la transacción.

Se debe tener en cuenta que los bonos no son transferibles pero si pueden usarse dentro del núcleo familiar, es decir que si un integrante del núcleo compra un bono puede ser utilizado por otra persona que tenga su mismo número de afiliado principal. Recordar que los bonos solo pueden ser utilizados para el plan que tenía asignado el afiliado en el momento que realizó la compra, es decir, que si luego cambia de plan, sea por uno más alto o uno más bajo, dichos bonos no podrán ser utilizados por él.

En ambos bonos debe figurar para qué plan puede utilizarse, además de permitir guardar el número del afiliado que lo utilizó.

Si se presenta el caso de que administrativo realiza la compra de bonos en nombre de un afiliado (por mostrador, el afiliado no compra el bono desde su casa y lo hace en la clínica) será necesario que ingrese el número del afiliado que realizó la compra en cuestión, pudiendo utilizar él o algún integrante de grupo familiar.

Para todos los casos de compra se tendrá que validar que el afiliado se encuentre activo.

## 10. Pedido de turno

Esta funcionalidad es la encargada de registrar el pedido de un turno para una atención médica por parte de un afiliado con algún profesional de la clínica en función de la agenda de este último.

Para ello se deberá determinar el profesional con el cual el afiliado se quiere atender, permitiendo refinar su búsqueda por especialidad.

Una vez determinado el profesional será necesario que se indique que fechas posee disponible para la atención (actual y futuras según la fecha del archivo config que determina la fecha del sistema) junto al día de la semana que le corresponde.

Bajo ningún punto de vista se permiten sobre-turnos o asignaciones fuera de las estipuladas por la agenda del profesional.

Una vez que se concertó la fecha se deberá informar los horarios disponibles para la misma el sistema generará un nro de turno único para esta futura atención.

## 11. Registro de Llegada para atención médica

Funcionalidad utilizada por un administrativo para registrar la llegada de un paciente a la clínica y efectivizar su turno con un especialista para convertirse en una consulta médica.

Para generar la consulta médica, será necesario que el afiliado indique al administrativo con que profesional tenía turno, pudiéndose refinar la búsqueda por especialidad, porque como ya es sabido, el paciente puede que no recuerde el nombre del profesional. Luego de determinado el profesional se cargaran todos los turnos disponibles para el día de la atención y mediante el ingreso del número de afiliado se validará si el dicho afiliado tenía turno con el profesional en cuestión.

Antes de finalizar este proceso que deja constancia de asistencia (de manera parcial) a un turno pedido por el afiliado, será necesario que se ingrese un bono consulta, se debe registrar en que consulta fue utilizado a modo de que dicho bono no sea nuevamente utilizado. Si por algún motivo de fuerza mayor el afiliado se retiró de la clínica, luego de que sea confirmada su asistencia, el bono consulta se tomará como consumido. Como dicha situación también puede darse para un médico, para simplificar la solución, suponemos que una vez que el médico está atendiendo, el mismo no se retira de la clínica y finaliza su jornada laboral en forma completa.

El proceso de registro de llegada tiene que persistir la fecha y hora en el que se produce la llegada a la clínica por parte del afiliado.

Se debe considerar que el paciente llegue siempre de manera puntual o anticipada, si llegase a dar el caso de que un paciente se retrasa, ya sea por un segundo, el turno está perdido. Es obligación de afiliando presentarse en recepción por lo menos 15 minutos antes del turno que tiene asignado y se garantiza que si se llega con dicha antelación, va a ser recepcionado por un administrativo para que no pierda el turno.

La única manera que existe de que un bono consulta pueda ser utilizado nuevamente se da solamente si el afiliado cancela su turno o el profesional cancela su día atención.

## 12. [Registro de resultado para atención médica](#)

Esta funcionalidad es utilizada por el profesional para registrar el diagnóstico propio de la atención.

Antes de ingresar el diagnóstico, será necesario que el profesional indique si la atención verdaderamente ocurrió junto con la fecha y hora del dicho acontecimiento.

Luego que el profesional examine al paciente, este cerrará la consulta ingresando cuales fueron los síntomas que produjeron la atención junto con el diagnóstico pertinente.

Tener en cuenta que toda atención puede generar o no una receta médica, ver punto 14 del TP.

## 13. [Cancelar atención médica](#)

La cancelación de una atención médica puede dar de dos formas:

- Cancelada por el paciente
- Cancelada por el médico

En caso de que la cancelación sea por parte del paciente, la misma deberá realizarse con un día de antelación. No puede cancelar un turno el mismo día de la atención se presupone que el paciente asistirá a su compromiso.

En caso de que la cancelación sea por parte del médico, tendrá la misma validación que para los pacientes.

Un paciente puede cancelar aquellos turnos que le fueron asignados.

Un médico puede cancelar un día particular de su agenda o un período determinado. Queda a criterio de los alumnos que se hace con los turnos cancelados por parte del médico, buscar un mismo especialista disponible o cancelarlos directamente.

Toda cancelación deberá tener un tipo de cancelación y luego un detalle más explicativo del porqué de la cancelación (motivo). Los tipos de cancelación quedan a criterio de los alumnos.

## 14. [Receta médica](#)

Esta funcionalidad se deriva de una consulta médica en función del resultado de la misma, para ser consumida en la farmacia de la clínica.

Básicamente para la confección de una receta médica se utiliza un bono farmacia, en cual solo pueden recetarse como máximo 5 medicamentos, en caso de que el profesional necesite indicar más medicamentos, se deberá utilizar otro bono farmacia.

Por cada medicamento recetado será necesario que se indique la cantidad respectiva junto con su aclaración en letras (que se ingresará automáticamente en función de la cantidad), no podrán indicarse más de 3 unidades por medicamento recetado, y un mismo medicamento no puede recetarse más de una vez por receta.

Es importante recordar que el bono no debe estar vencido para su utilización y que debe guardarse en el bono la fecha de la prescripción médica de los medicamentos.

No existe posibilidad de modificación y/o eliminación de una receta una vez que la misma se encuentra confeccionada

## 15. Listado Estadístico

Esta funcionalidad nos debe permitir consultar el TOP 5 de:

- Top 5 de las especialidades que más se registraron cancelaciones, tanto de pacientes como de profesionales
- Top 5 de la cantidad total de bonos farmacia vencidos por afiliado.
- Top 5 de las especialidades de médicos con más bonos de farmacia recetados.
- Top 10 de los afiliados que utilizaron bonos que ellos mismo no compraron.

Dichas consultas son a nivel semestral pero con desagregación mensual para lo cual la pantalla debe permitirnos la selección del semestre a consultar.

El listado se debe ordenar en forma descendente en función de las cantidades según corresponda.

Además de ingresar el año a consultar, el sistema nos debe permitir seleccionar que tipo de listado se quiere visualizar.

Cabe aclarar que los campos a visualizar en la tabla del listado para las consultas no son los mismos, y al momento de seleccionar un tipo solo deben visualizarse las columnas pertinentes al tipo de listado elegido.

Las columnas del listado para cada una de las consultas quedan a cargo del alumno y dichas columnas deben ser lo suficientemente descriptivas para poder brindar un informe detallado.

## **Implementación**

### **General**

El alumno deberá desarrollar dos componentes: un script de base de datos SQL Server y una aplicación Desktop C#.

A continuación se detalla la implementación de cada componente:

### **Base de Datos**

El alumno debe instalar el motor de base de datos SQL Server 2008 con las siguientes consideraciones:

- El nombre de la instancia del motor de base de datos a instalar debe llamarse “SQLSERVER2008”. No utilizar el nombre “Default” para la instancia. Instalar como instancia con nombre (“Named Instance”).
- La autenticación debe ser por “Modo Mixto”.
- El usuario administrador de la base de datos deberá tener la siguiente configuración:
  - Username: “sa”
  - Password: “gestiondedatos”

Una vez instalado el motor de base de datos se deberán instalar las herramientas cliente de trabajo: “Microsoft SQL Server Management Studio Express” para SQL Server 2008. Ejecutar esta aplicación e ingresar los datos del usuario “sa” creado anteriormente (en modo “Autenticación de SQL Server”).

Dentro del “Management Studio” crear una nueva base de datos con los parámetros default y nombre de base “GD2C2013”.

Crear un nuevo “Inicio de Sesión”, desde el ítem “Seguridad” perteneciente al servidor de Base de Datos general. El inicio de sesión debe poseer las siguientes características:

- Solapa “General”:
  - Nombre de inicio de sesión: “gd”
  - Autenticación de SQL Server
  - Contraseña: “gd2013”
  - Base de Datos Predeterminada: GD2C2013.
  - El resto de los parámetros respetar sus valores default.
- Solapa “Funciones del Servidor”:
  - Seleccionar “sysadmin”
- Solapa “Asignación de Usuarios”:
  - Seleccionar asignar a “GD2C2013”
- Para el resto de los parámetros respetar sus valores default.

Salir del “Management Studio” como usuario “sa” y volver a ingresar con el nuevo usuario “gd” creado. Es probable que informe que la contraseña ha caducado. Cambiar la contraseña ingresando exactamente la misma que antes: “gd2013”.

Una vez que tenemos la base de datos creada y configurada con el usuario, necesitamos ejecutar dos scripts. Para ello debemos ejecutar un comando de consola de SQL Server llamada “sqlcmd”. Este comando debe ejecutar en orden los siguientes dos archivos:

- gd\_esquema.Schema.sql: Este archivo genera un esquema llamado “gd\_esquema” dentro de la base de datos y lo asigna al usuario “gd”.
- gd\_esquema.Maestra.Table.sql: Este archivo crea la tabla principal del trabajo práctico y la carga con los datos correspondientes. El archivo posee un volumen significativo y no puede ser ejecutado desde el “Management Studio”.

La cátedra provee un archivo BATCH para ejecutar esta operación, denominado “EjecutarScriptTablaMaestra.bat”. Haciendo doble clic sobre el mismo se ejecutan ambos archivos (“gd\_esquema.Schema.sql” y “gd\_esquema.Maestra.Table.sql”) a través del modo consola. El Script necesita aproximadamente 40 minutos para finalizar su ejecución.

```
sqlcmd -S <Servidor\Instancia> -U <Nombre_de_usuario> -P <Password> -i  
<Nombre_del_archivo1>,<Nombre_del_archivo2> -a 32767
```

Ejemplo:

```
sqlcmd -S localhost\SQLSERVER2008 -U gd -P gd2012 -i  
gd_esquema.Schema.sql,gd_esquema.Maestra.Table.sql -a 32767 -o  
resultado_output.txt
```

Luego de cargados todos los datos de la tabla maestra, el alumno deberá crear su propio esquema dentro de la base de datos. El nombre del esquema deberá ser igual al nombre del grupo registrado en la materia (el proceso de registración se explica más adelante). El nombre del esquema debe ser en mayúsculas, sin espacios y separado por guiones bajos. Ejemplo “Los mejores” debe ser “LOS\_MEJORES”.

Todas las tablas, stored procedures, vistas, triggers y otros objetos de base de datos nuevos que cree el alumno deberán pertenecer a este esquema creado. Sin la solución entregada posee objetos de base de datos por fuera del esquema con el nombre del grupo, el TP será rechazado sin evaluar su funcionalidad.

Con esta configuración el alumno está listo para empezar la implementación de la parte de base de datos.

## **Aplicación Desktop**

La cátedra provee una aplicación Desktop en C#, a modo de template, sobre la cual se debe desarrollar la aplicación del Trabajo Práctico.

Para ejecutar esta aplicación es necesario instalar Visual Studio 2008 con el Framework de .NET 3.5. La versión Express posee la funcionalidad necesaria como para desarrollar el Trabajo Práctico.

La aplicación template se denomina “ClinicaDesktop”. Cuenta con un formulario principal, una barra de menú y un formulario para cada funcionalidad visual que hay que implementar en el trabajo. El alumno debe depositar su código respetando esta estructura.

Más allá de estas indicaciones, el alumno puede modificar a su criterio la aplicación template. Ante cualquier consulta sobre lo que se puede modificar consultar al grupo de Google de la materia.

La aplicación Desktop deberá conectarse a la base de datos con los siguientes parámetros:

- Origen de datos: Microsoft SQL Server (SqlClient)
- localhost\SQLSERVER2008
- Utilizar autenticación de SQL Server:
  - Nombre de Usuario: gd
  - Password: gd2013
- Nombre de la base de datos: GD2C2013

La aplicación siempre debe conectarse a localhost, pero al menos una vez debe ser probado en una red, dado que la corrección se realizara en red. En caso de que el alumno se conecte a otra dirección, deberá cambiarlo a la hora de entregar su TP para que sea corregido.

La aplicación deberá contar con un usuario de sistema ya creado (además del set que fue pedido en puntos anteriores) y listo para ser utilizado, con las siguientes características:

- Username: admin
- Password: w23e
- Rol:
  - Nombre: Administrador General
  - Funcionalidades: todas las existentes

Este usuario de aplicación debe ser generado en forma automática dentro del archivo “script\_creacion\_inicial.sql” y quedar listo para ser utilizado por la aplicación Desktop.



## Restricciones de la solución

El lenguaje de programación utilizado deberá ser únicamente C# utilizando el Framework .NET 3.5. Cualquier otra implementación que no halla sido desarrollado en éste lenguaje será rechazada, sin excepción.

El entorno de desarrollo debe ser Microsoft Visual Studio 2008 o Microsoft Visual C# Studio Express 2008. No podrá ser utilizada la reciente versión 2010.

El motor de base de datos deberá ser Microsoft SQL Server 2008. Tanto la versión Express como la full sirven para realizar el trabajo. No podrá ser utilizada la reciente versión 2010.

No podrá utilizarse ninguna herramienta auxiliar que ayude a realizar la migración de datos. Tampoco podrá desarrollarse una aplicación personalizada para la migración de datos. La misma deberá ser efectuada en código T-SQL en el archivo de script “script\_creacion\_inicial.sql”.

## Condiciones de aprobación

### Testing

El alumno deberá entregar dos componentes:

- Un único script de base de datos (script\_creacion\_inicial.sql) con todo lo necesario para crear su modelo y cargarlo con datos.
- La aplicación C# “ClinicaDesktop” con la funcionalidad pedida.

La cátedra probará el Trabajo Práctico en el siguiente orden:

1. Disponer de una base de datos limpia igual a la original entregada a los alumnos.
2. Ejecutar el archivo script\_creacion\_inicial.sql. Este archivo debe tener absolutamente todo lo necesario para crear y cargar el modelo de datos. Toda la ejecución debe realizarse en orden y sin ningún tipo de error ni warning.
3. Se ejecuta la aplicación Desktop y se prueban las funcionalidades pedidas.

El archivo “script\_creacion\_inicial.sql” debe contener todo lo necesario para crear el modelo de datos y cargarlo. Si el alumno utilizó alguna herramienta auxiliar o programa customizado, el mismo no será utilizado por la cátedra.

**Si el script de base de datos ejecuta con errores, el trabajo práctico será rechazado sin continuar su evaluación.**

Todos los objetos de base de datos nuevos creados por el usuario deben pertenecer a un esquema de base de datos creado con el nombre del grupo. Si esta restricción no se cumple el trabajo práctico será rechazado sin continuar su evaluación.

## **Modelo de Datos**

El modelo de datos creado por el alumno deberá respetar las buenas prácticas de programación y diseño de bases de datos explicados durante la cursada de la materia.

También deberán ser considerados criterios de performance a la hora de crear relaciones e índices en las tablas.

## **Consultas SQL**

Todas las consultas SQL que haga la aplicación serán evaluadas de acuerdo al standard de programación SQL explicados en clase. La performance de las mismas será tomada en cuenta a la hora de fijar la nota.

## **Respetar Guía de ABMs**

Todo el código y las pantallas creadas en la aplicación Desktop deberá respetar a la perfección los lineamientos especificados en el documento “Guía de ABMs”. Aquellos TPs que no respeten las indicaciones en forma total serán rechazados, por más que cumplan la funcionalidad pedida.

## **Aplicación Desktop**

La calidad y orden del código fuente será tomada en cuenta a la hora de fijar la nota. Es obligatorio que existan comentarios de código en todas las secciones principales de implementación.

Deberán crearse componentes de código reusable para aquellas porciones de código ejecutadas en muchas secciones de la aplicación. Todo tipo de configuración o parametrización de la aplicación deberá estar centralizado en un solo punto. Aquellos TPs que no respeten estos puntos mencionados serán rechazados, sin continuar su evaluación (ej: los parámetros de conexión a la base de datos).

## **Fecha de entrega y condiciones**

Existe una sola fecha de entrega posible para el TP, pudiendo presentarse hasta 2 veces más (entregas que no tienen fecha fija).

La cantidad de funcionalidad de cada entrega no varía.

- *Entrega unica:*
  - *Día:* 15/11/013 hasta las 12:00 del mediodía (GMT -3:00 Buenos Aires). Los TPs entregados ese día, después de las 12:00 del mediodía se consideran que no son entregados en fecha y le restan 2 oportunidades más de entrega, como se menciona anteriormente. Se considera que la fecha última de entrega sea el día 15/12/2013.

En función de la cantidad de trabajos entregados, la corrección puede demorar hasta 7 días hasta las 12:00 del mediodía (GMT -3:00 Buenos Aires).

## **Sobre los grupos**

Deberán estar compuestos de no más de cuatro integrantes. Cada grupo debe tener un representante que será el único que podrá enviar mails con el TP para su corrección. Los grupos pueden estar compuestos por alumnos de distinto curso. Los alumnos deben registrar su grupo en un sitio de registración especial, especificando un nombre único que identifique al grupo. La URL del sitio de registración es la siguiente:

<https://spreadsheets0.google.com/viewform?formkey=dG16aEltMHc1X2hPN3U2YTVoVGxfeUE6MA>

Al registrarse es necesario especificar un nombre de grupo. El nombre debe ser en mayúsculas, sin espacios y separado por guiones bajos. Ejemplo “Los mejores” debe ser “LOS\_MEJORES”.

Luego, el 15/10/2013 se enviarán los mail correspondientes con la confirmación de los grupos y se les asignará un número de grupo además del nombre que debidamente eligieron. Luego de esa fecha, la cátedra enviará al grupo **OFICIAL** la conformación de los mismos que será inalterable hasta la finalización del cuatrimestre. Cualquier cambio de integrantes, sea por el motivo que fuese, deberá realizarse antes de esa fecha. No aceptándose ninguna modificación pasada dicha fecha. Es obligación de los alumnos ingresar al grupo de la cátedra para obtener dicha información.

## **Ayuda y contacto**

El sitio oficial de la materia es el siguiente:

<https://sites.google.com/site/gestiondedatosutn>

También existe un grupo de Google en donde se podrán plantear dudas sobre el TP. Su dirección es la siguiente:

<http://groups.google.com/group/gestiondedatos>

Todos los mensajes referentes al trabajo práctico deberán contener la etiqueta [TP] antes del asunto. Ej: “[TP] consulta sobre base de datos”.

Es obligación del alumno revisar el grupo periódicamente y mantenerse informado sobre actualizaciones, cambios de consignas, modificaciones del programa, cambios de fecha, etc.

La cátedra no asigna ayudantes específicos a cada grupo. Todas las consultas deberán hacerse a través del grupo de Google.

Es obligación de los alumnos ingresar periódicamente al grupo para informarse sobre cuestiones del TP. El grupo OFICIAL es el único medio de comunicación con los alumnos sobre cuestiones del trabajo práctico. La cátedra no se hará responsable si existen grupo paralelos y que no sea el que se detalló en el siguiente enunciado.

Cualquier tipo de información sobre el trabajo práctico que haya sido brindada por los docentes, tendrá que ser validada con los ayudantes ya sea el Ing. López Matias Miguel o el Ing Ariosti Maximiliano.

A lo largo de la cursada pueden ir surgiendo dudas particulares sobre el Trabajo Práctico que sean útiles para el resto de los alumnos. Para ello la cátedra cuenta con un documento denominado “Apéndice del Enunciado” en el cuál se agregan consideraciones generales de manera online. Su dirección es la siguiente:

<https://docs.google.com/document/d/13aTC4-yaN3TtaiMlGo7pIlw9p9tkl3h3G9dvHrZcBrk/edit>

Inicialmente el Apéndice se encuentra vacío. A medida que vayan surgiendo dudas sobre el desarrollo del Trabajo Práctico, la cátedra evaluará agregar consideraciones generales al documento. Es obligación del alumno revisar este documento periódicamente.

## **Donde aprender C#**

Si bien para resolver el TP solo se necesita conocer una pequeña parte de la totalidad del lenguaje C#, es recomendable aprender los conceptos básicos mediante algún libro o tutorial. Recomendamos el siguiente tutorial:

<http://www.devjoker.com/asp/~gru/Tutorial-C/TUCS/Tutorial-C.aspx>

En cuanto a las colecciones que posee .Net, recomendamos la siguiente documentación:

Documentación de MSDN en español

<http://msdn2.microsoft.com/es-es/library/7v3x785f%28VS.80%29.aspx>

## **Sobre la elección de C#**

El lenguaje C, tradicionalmente usado en las cátedras de la facultad, ha demostrado tener cierta dificultad a lo hora de su uso. El uso de punteros y procesamiento de cadenas muchas veces resulta complicado, sin mencionar la dificultad de encontrar un error en tiempo de ejecución. También hemos notado que se invierte mucho tiempo tratando de crear interfaces amigables mediante consola de texto.

Es por esto que creemos que C# al igual que otros lenguajes de última generación, como Java, permiten invertir más tiempo en cuestiones algorítmicas y de estructura de datos, dejando de lado las cuestiones de sintaxis propias del lenguaje C.

Por nombrar algunas ventajas de C#:

- Facilidad en la depuración en tiempo de ejecución: Nos permite inspeccionar el valor de las variables durante la ejecución del programa, incluso visualizar estructuras de datos recursivas.
- Las sintaxis está totalmente normalizada.
- El IDE permite autocompletar código.
- Provee métodos simples para el manejo de E/S.

Podemos decir que gran parte de la eficiencia de un programa depende no del lenguaje en el cual es implementado, sino de las estructuras de datos y algoritmos elegidos para resolverlo.

Por último consideramos que el paradigma orientado a objetos puede brindarnos muchas ventajas que a esta altura ya resultan evidentes y al mismo tiempo permite aplicar los mismos esquemas algorítmicos que los lenguajes estructurados.

## Obtención de herramientas

El TP puede ser desarrollado con dos versiones del IDE Microsoft Visual Studio 2008. No puede utilizarse la versión 2010:

1. **Microsoft Visual Studio Professional 2008:** esta versión puede ser obtenida con licencia universitaria completa, gracias a un convenio de Microsoft con la UTN. Dirigirse al laboratorio de Microsoft ubicado en la sede Medrano (planta baja, hacia la derecha, mirando desde la entrada de la facultad hacia adentro). La versión ofrecida es en español y contiene la ayuda completa (MSDN). Para encargarla es necesario llevar un DVD y completar un formulario. Es posible que el programa halla que encargarlo y pasarlo a buscar otro día.
2. **Microsoft Visual C# 2008 Express Edition:** Existe una versión gratuita del IDE llamada *Visual C# 2008 Express Edition*, la cual posee todas las herramientas necesarias para realizar el TP. Esta se encuentra disponible en:

<http://download.microsoft.com/download/8/B/5/8B5804AD-4990-40D0-A6AA-CE894CBBB3DC/VS2008ExpressENUX1397868.iso>

El motor de base de datos a utilizar es SQL Server 2008 Express. Puede ser descargado de la siguiente dirección:

<http://www.microsoft.com/es-ar/download/details.aspx?id=1695>

Es necesario descargar e instalar dos componentes:

- Install Microsoft SQL Server 2008 Express Edition
- SQL Server Management Studio Express

## Formato de entrega

## Lugar de envío

La entrega debe realizarse por mail el antes de las fechas estipuladas en el documento de enunciado

La dirección del mail es:

[gestiondedatos.entregas@gmail.com](mailto:gestiondedatos.entregas@gmail.com)

El asunto del mail debe cumplir con el siguiente formato:

TP2C2013<b><curso><b><nombreGrupo><Nro de grupo>

<b>: espacio en blanco

Ejemplos:

TP2C2013 k9999 LOS\_MEJORES 10

(Respetar los 2 espacios en blanco existentes)

Se debe adjuntar el trabajo práctico en un archivo del tipo zip con el mismo nombre que el asunto del mail.

**Por cuestiones de seguridad Gmail rechaza todos los adjuntos que contengan archivos zip con .exe y .dll en su interior, por lo que es necesario renombrar la extensión .zip a .zip123.**

Por ejemplo:

TP2C2013 k9999 LOS\_MEJORES 10.zip123

(Respetar los 2 espacios en blanco existentes)

**No enviar adjuntos de más de 20 MB. La casilla de mail rechazará mails que superen esta restricción.**

El cuerpo del mail debe contener lo siguiente:

Grupo:

Curso:

Integrantes: <apellido>, <nombres> - <legajo>

*Nota: En caso de que haya integrantes de cursos distintos, se debe poner el curso de la persona elegida como representante*

*En caso de que algún alumno del grupo haya dejado de cursar o se halla cambiado de grupo, deberá ser aclarado en el mail de la entrega del TP.*

*Solo debe enviarse la entrega desde el mail del representante del grupo.*

Los alumnos deberán registrar su grupo en la dirección mencionada anteriormente. No se aceptarán TPs que no estén registrados.

## **Estructura del archivo zip**

El archivo zip (.zip123) debe contener la siguiente estructura de directorios:

```
⇒ \  
    ⇒ Readme.txt  
    ⇒ Estrategia.pdf  
    ⇒ \src  
        ⇒ Solución entera de Visual Studio de “ClinicaDesktop”  
    ⇒ \data  
        ⇒ Archivo de script de base de datos “script_creación_inicial.sql”.
```

### **Readme.txt:**

Es un archivo de texto plano con los siguientes datos:

- Curso
- Número de grupo
- Nombre y legajo de todos los integrantes
- Email del integrante responsable del grupo.

### **Estrategia.pdf:**

Archivo PDF en donde se deberá explicar en forma detallada y extensa la estrategia utilizada para desarrollar el TP. Debe incluir una descripción de todas las estructuras de datos relevantes utilizadas en el algoritmo, explicando la razón de la elección de dichas estructuras. Cualquier consideración tomada o asumida deberá ser aclarada en este documento.

Se debe incluir un DER del modelo de datos creado con una explicación detallada de cada entidad, relaciones, claves primarias y foráneas, índices, stored procedures, triggers, vistas, etc.

Sin este archivo y un DER LEGIBLE de la base la entrega no será tomada como válida. Se recomienda que el DER se entregue en horizontalmente. En caso que no entre en el ancho de la hoja, SE DEBERÁ ENTREGAR UNA IMAGEN, EN EL FORMATO QUE SEA, DE MANERA QUE SEA LEGIBLE.

SI EL DER NO ES LEGIBLE, NO SE CORRIGE EL TP

### **\src:**

Dentro de este directorio se encuentra la solución entera de Visual Studio del proyecto “ClinicaDesktop”. Evitar enviar archivos de SVN, CVS u otros. No enviar archivos ejecutables ni DLLs. Limpiar la solución desde Visual Studio (Proyecto => Limpiar Solución) antes de enviarla.

Además dentro del source deberá existir un archivo de configuración en donde se encuentren los parámetros de conexión a la base de datos, **la fecha que tomará el sistema para funcionar** (se utiliza este criterio para simplificar al alumno el manejo de las fechas y así evitar el cambio de fechas del sistema operativo).

Sin ese archivo de configuración la entrega no será tomada como válida.

### **\data:**

Archivo “script\_creación\_inicial.sql” con toda la creación del modelo de datos. El archivo debe poder ejecutar perfectamente de una sola vez, sin ningún tipo de error. Todas las sentencias deben estar perfectamente ordenadas para ejecutar correctamente. Cada sentencia debe estar comentada explicando su intención.

Cualquier TP entregado que no cumpla con alguno de los requisitos mencionados en este documento, será rechazado sin ser evaluado, perdiendo una oportunidad de reentrega.

## **Decálogo de aprobación Gestión de Datos**

1. No inscribirse en el grupo
2. No entregar el DER del modelado, y en caso de entregarlo, que el mismo sea legible
3. No consultar periódicamente el grupo de la materia
4. Entregar trabajos prácticos anteriores
5. Entregar el script de migración y/o solución de C# con errores
6. Entregar ABMs sin validaciones propias de un AMB, por ej. permitir alta de registros duplicados.
7. Hacer mal uso de los formatos de fecha. ( se recomienda convert ) y/o no usar el archivo config que se les pide.
8. Entregar el TP pasada la fecha límite para la recepción para su corrección
9. Realizar una única entrega en una fecha cercana a la fecha límite sin la posibilidad de un feedback y si aún así deciden realizar una única entrega asumen el riesgo de Aprobación o Desaprobación
10. Si hay dudas, volver a leer los puntos anteriores o consultarlos con los ayudantes.